

Программный пакет генерации целочисленных преобразований

Григор А.Петросян

Институт проблем информатики и автоматизации НАН РА
e-mail spl@ipia.sci.am

Аннотация

Пакет даст возможность в зависимости от диапазона представления чисел данной системы, генерировать матрицы целочисленных преобразования Фурье, DST-1-4, DST-1-4 и Хаара порядка 2^n .

1. Введение

В задачах обработки изображений и сигналов широко применяются ортогональные преобразования (Фурье, Адамара, синусоидальные преобразования), что обусловлено их свойством накопления энергии в спектральной области и наличием быстродействующих алгоритмов вычислений. Из них самое широкое применение получило дискретное косинусное преобразование второго типа, используемое в стандартах JPEG (ISO 10918-1) MPEG (ISO/IEC 11172) [1].

В работе представлено краткое описание пакета, генерирующего целочисленные преобразования Фурье, четырех типов косинусных, четырех типов синусных, а также целочисленного преобразования Хаара порядка 2^n . В первой части представлены теоретические основы чисел с фиксированными и плавающими запятыми, микропроцессоры работающие с этими числами, а также их преимущества и недостатки. Во второй части обсуждаются общие основы генерации целочисленных преобразований. В третьей части приводятся принципы применения SIMD (single-instruction, multiple-data) архитектуры [2] в реализации целых преобразований. В четвертой части представлены экспериментальные результаты применения целочисленного косинусного преобразования и пример генерирования целочисленного косинусного преобразования.

2. Системы с плавающей и фиксированной запятой

В системах обработки цифровых сигналов (СОЦС) в основном используется, или арифметика с плавающей запятой (floating point arithmetic), или арифметика с фиксированной запятой (fixed point arithmetic). В СОЦС более распространена арифметика с фиксированной запятой, так как она более быстрая с точки зрения реализации и более дешевая [3]. Однако отметим, что диапазон чисел ограничен, и по этой

причине при расчетах приходится сталкиваться с проблемой переполнения. Чтобы избежать проблем, возникающих при арифметических операциях приходится масштабировать результаты, что, в свою очередь, является первопричиной для понижения отношений сигнал-шум и влияет на производительность СОПС.

Как уже было сказано, по сравнению со своими классическими аналогами целочисленные преобразования выделяются быстродействием и экономичным расходом памяти, при этом качественно почти не уступая последним. В среде обработки сигналов входные данные в основном целочисленные, занимают область памяти в один байт, в том случае, что во время классических преобразований имеем дело с числами с плавающей запятой одинарной точности (float - 4 байт) или двойной точности (double - 8 байт). По этой причине приходится перейти на арифметику с плавающей запятой, что в основном замедляет процесс обработки.

Арифметика с плавающей запятой предпочтительнее, когда диапазон переменных используемых в СОПС может значительно изменяться [4]. Она позволяет использовать намного более широкую область представления чисел, и про проблемы связанные с переполнением можно забыть. Также было замечено, что реализация арифметики с плавающей запятой является более дорогостоящей и частично более медленной, так как в последнее время широкое распространение получают быстродействующие процессоры, предназначенные для обработки сигналов, которые снабжены вложенным процессором, предназначенным для арифметики с плавающей запятой (например: Texas Instruments TMS32C30) [3]. Заметим также, что хотя в последнее время разницы в цене между процессорами с плавающей и фиксированной запятыми все уменьшается, тем не менее потребность в быстродействии и экономии памяти является важным фактором для использования процессоров с фиксированной запятой и разработки для них новых алгоритмов.

Как пример можно отметить вложенные процессоры (Floating Point Unit - FPU) на процессорах PENTIUM и AMD для реализации операций с плавающей запятой, получившие широкое распространение [2]. FPU наделенное SIMD архитектурой предназначено также для поточной обработки данных, и в его каждом 64 битном mpx или 128 битном xmm регистре можно одной командой обработать соответственно 8 и 16 байтных массивов данных, вместо 2 и 4 float или 1 и 2 double чисел с плавающей запятой.

В некоторых системах ЦОС числа представляются в виде дробей. Каждое число принадлежит интервалу $[-1, 1 - 2^{-(B-1)}]$, где B - необходимое число битов для представления числа. В СОПС стандарт Q является применяемым стандартом для представления чисел. Например в стандарте Q15 используются 16 бит (1 бит является знаковым и 15 бит содержат дробную часть). В таких системах представление положительных чисел не отличается от классического бинарного представления. Отрицательное число получается бинарным дополнением всех его битов и прибавлением 1 к младшему биту. Например, для представления числа $-3/8$ необходимо число $(3=0011)$ представить его бинарным дополнением и прибавить единицу $(1100+0001=1101)$.

Если в системе бинарного дополнения для представления каждого числа необходимы B битов, то можно представить самое большее 2^B чисел, при этом соседние числа будут смещены друг от друга на 2^{-B} . По этой причине целесообразно принять ту точность, при которой можно представить любое число, соответствующее десятичному числу. Для данной десятичной дроби, которая состоит из d чисел, точность равна $\pm 0,5 \times 10^{-d}$. Чтобы обеспечить ту же точность, необходимо: $0,5 \times 10^{-d} = 0,5 \times 2^{-B}$, т.е. $B = d \log_2 10 \approx 3,3d$ битов. Например, если хотим десятичное число 0,23456 представить с той же точностью, необходимо $3,3 \times 5 = 17$ бит.

Принимая во внимание вышесказанное, целесообразно генерировать матрицы преобразования в зависимости от возможности системы и необходимой точности. Ссылаясь на ITG пакет, необходимо сказать, что он генерирует матрицы целочисленных преобразований, приближая числа с плавающей запятой в зависимости от диапазона представления чисел системы или необходимой точности. Например, можно генерировать целочисленные матрицы преобразования для N битного процессора или целочисленные матрицы преобразования которые сохранят точность в M знаках после запятой.

3. Синтез целочисленных преобразований

Для получения целочисленных преобразований целесообразно пользоваться их быстрыми вариантами, которые используют меньше операций, что приводит к уменьшению погрешности. Одним из распространенных методов получения этих вариантов является факторизация матрицы преобразования с помощью разреженных и лифтинг матриц. Матрица N -ого порядка $A = (a_{n,m})$ называется лифтинг матрицей, если она удовлетворяет следующим условиям:

$$A_{m,n} = \begin{cases} s, & \text{если } n = i \text{ и } m = j, \\ 1, & \text{если } n = m, \\ 0, & \text{в ост. сл.} \end{cases} \quad \text{где } 0 \leq n, m \leq N-1, \quad s \neq 0.$$

Лифтинг шаг [5], который является произведением лифтинг матрицы и вектора, можно рассчитать используя всего одно сложение и одно произведение. Предположим x и y — векторы длины N , и $y = L_{i,j}(s)x$, тогда

$$y(i) = x(i) + sx(j) \text{ и } y(k) = x(k) \quad k \neq i.$$

Рассмотрим быстрые варианты преобразований применяемых в ITG пакете.

Для быстрого преобразования Фурье выбран алгоритм Кули-Тука [6], который требует $N/2 \log_2 N$ комплексных умножений и $N \log_2 N$ комплексных сложений. Каждый комплексный коэффициент может быть представлен в виде $a = c + js$. Пусть $x = x_r + ix_i$ — комплексное число, тогда $y = ax = (cx_r - sx_i) + j(cx_i + sx_r)$ или в матричной форме

$$y = \begin{bmatrix} 1 & j \\ s & c \end{bmatrix} \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} x_r \\ x_i \end{bmatrix} = \begin{bmatrix} 1 & j \\ s & c \end{bmatrix} R \begin{bmatrix} x_r \\ x_i \end{bmatrix} \quad \text{где } R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}.$$

Поскольку c и s — коэффициенты Фурье и $c^2 + s^2 = 1$, а R — ортонормальная матрица, то его можно факторизировать с помощью трех лифтинг матриц [7], каждая из которых далее может быть заменена своими целочисленными вариантами

$$R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} = \begin{bmatrix} 1 & \frac{c-1}{s} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ s & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{c-1}{s} \\ 0 & 1 \end{bmatrix}.$$

В пакете ITG для факторизации матриц синусоидальных преобразований [8] были использованы факторизационные формулы Вонга [4]. В ниже приведенных формулах $N = 2^n$, нижний индекс указывает на порядок матрицы, а верхний на тип преобразования. (C'_M и S'_M — матрицы дискретных косинусных и синусных преобразований порядка M типа i.)

$$\begin{aligned}
 C_{N+1}^I &= P_{N+1} \begin{bmatrix} C_{N/2+1}^I \\ \overline{C}_{N/2}^m \end{bmatrix} A_{N+1}, & S_{N-1}^I &= P_{N-1} \begin{bmatrix} S_{N/2}^{III} \\ \overline{S}_{N/2-1}^I \end{bmatrix} A_{N-1}, \\
 C_N^{II} &= P_N \begin{bmatrix} C_{N/2}^{II} \\ \overline{C}_{N/2}^{IV} \end{bmatrix} A_N, & S_N^{II} &= P_N \begin{bmatrix} S_{N/2}^{IV} \\ \overline{C}_{N/2}^{II} \end{bmatrix} A_N, \\
 C_N^{III} &= [C_N^{II}]^T = [C_N^{II}]^T = A_N \begin{bmatrix} C_{N/2}^{III} \\ \overline{C}_{N/2}^{IV} \end{bmatrix} [P_N]^T, & S_N^{III} &= A_N \begin{bmatrix} S_{N/2}^{IV} \\ \overline{C}_{N/2}^{III} \end{bmatrix} [P_N]^T, \\
 C_N^{IV} &= Q_N V_N (J) U_N (J-I) V_N (J-I) \dots U_N (I) V_N (I) H_N, & S_N^{IV} &= R_N C_N^{IV} D_N.
 \end{aligned}$$

Здесь P , A , Q , H , R и D разреженные матрицы, заполненные нулями и единицами. Две черточки над данной матрицей указывают на то, что строки и столбцы матрицы расположены с противоположной очередностью. V и U являются блочно-диагональными матрицами содержащими в себе повороты.

Каждый из этих поворотов может быть вычислен с помощью трех сложений и трех умножений

$$R = \begin{bmatrix} \cos \alpha & \sin \alpha \\ \sin \alpha & -\cos \alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha + \sin \alpha & 0 & 0 \\ 0 & -\cos \alpha + \sin \alpha & 0 \\ 0 & 0 & \sin \alpha \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & -1 \end{bmatrix}.$$

Для факторизации матрицы преобразований Хаара [6] применяется факторизация предложенная Агаяном и Геворкяном, в результате которой получают матрицы перемещения и блочно-диагональные матрицы, которые содержат повороты типа $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. Легко заметить, что здесь также применима вышеупомянутая факторизация.

4. Применение SIMD технологии

Рассмотрим преимущества целочисленных преобразований с применением SIMD технологии. Как уже было отмечено выше, преимуществом целочисленных преобразований было экономичное расходование памяти и быстродействие, которое очевидным образом проявляется при применении SIMD технологии. Рассмотрим пример матрицы, которая получается при факторизации матрицы дискретного косинусного преобразования четвертого типа порядка 8.

$$U_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

С помощью mmx или xmm регистров можно умножить матрицу U сразу на четыре или восемь векторов соответственно, как показано на рис. 1, поскольку каждым mmx или xmm

регистром с помощью одной команды обрабатывается 4 или 8 short (2 байт) типа переменных. А при работе с числами с плавающей запятой возможна обработка соответственно 2 и 4 float и 1 и 2 double типа переменных.

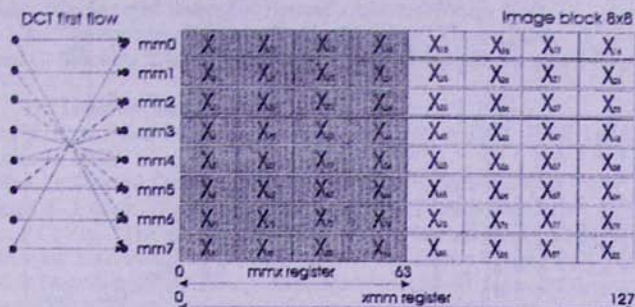


Рис. 1: Применение SIMD технологии

5. Результаты эксперимента

Опыты проведенные на изображении LENA.BMP(256x256) свидетельствуют о том, что целочисленное дискретное косинусное преобразование второго типа своими качественными свойствами находясь близко к классическому преобразованию, с точки зрения быстродействия значительно превышает последнее. Ниже приведены результаты сравнения преобразований второго типа целочисленного и классического дискретного косинусного преобразования (таблица 1,2) [8]. Опыты были проведены на процессоре AMD Athlon XP 1700+.

Таблица 1. Время выполнения преобразований для одного блока размером 8x8

	DCT-II (double precision)	DCT-II with MMX (single precision)	DCT-II with MMX (double precision)
Processor Ticks	4046	580	427

Таблица 2. Коэффициенты сжатия и потери качества для Y компонента

	DCT-II (double precision)	DCT-II with MMX (single precision)	DCT-II with MMX (double precision)
MSE	5.79	5.79	6.07
PSNR	32.88	32.88	32.32
Comp. Ratio	4.78	4.78	4.54



(a)



(b)



(c)

Рис. 2 (а)- начальное изображение, (b)- DCT-II, (c) IntDCT-II

Ниже представлена матрица преобразования C_8^{IV} , сгенерированная пакетом ITG.

$$C_8^{IV} = \sqrt{\frac{2}{8}} Q_8 I_8 E_8 U_8^2 \begin{bmatrix} I_4 & & & \\ & E_4 & & \\ & & I_4 & \\ & & & E_4 \end{bmatrix} U_8^1 \begin{bmatrix} I_2 & & & \\ & E_2 & & \\ & & I_2 & \\ & & & E_2 \end{bmatrix} H_8$$

где

$$Q_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, E_8 = \begin{bmatrix} C_{1/32} & S_{1/32} & & & & & & \\ S_{1/32} & -C_{1/32} & & & & & & \\ & & C_{3/32} & S_{3/32} & & & & \\ & & S_{3/32} & -C_{3/32} & & & & \\ & & & & C_{5/32} & S_{5/32} & & \\ & & & & S_{5/32} & -C_{5/32} & & \\ & & & & & & C_{7/32} & S_{7/32} \\ & & & & & & S_{7/32} & -C_{7/32} \end{bmatrix}, U_8^2 = \begin{bmatrix} I_4 & & & \\ & I_4 & & \\ & & I_4 & \\ & & & I_4 \end{bmatrix},$$

$$E_4 = \begin{bmatrix} C_{1/8} & S_{1/8} & & \\ S_{1/8} & -C_{1/8} & & \\ & & C_{3/8} & S_{3/8} \\ & & S_{3/8} & -C_{3/8} \end{bmatrix}, U_8^1 = \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \\ & I_2 & I_2 \\ & I_2 & -I_2 \end{bmatrix}, E_2 = \begin{bmatrix} C_{1/4} & S_{1/4} \\ S_{1/4} & -C_{1/4} \end{bmatrix}, H_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix},$$

где $C_i = \cos(i\pi)$ и $S_i = \sin(i\pi)$.

Легко заметить, что матрица преобразования C_8^{IV} выражена с помощью разреженных целых матриц и матриц, содержащих повороты. Повороты, как было уже сказано, представляются тремя матрицами, где и происходит их приближение (аппроксимация).

Например

$$R_{1/32} = \begin{bmatrix} \cos 1/32 & \sin 1/32 \\ \sin 1/32 & -\cos 1/32 \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 220/256 & 0 \\ 0 & 0 & 25/256 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & -1 \end{bmatrix}.$$

6. Заключение

Разработанная система дает возможность генерировать целочисленные преобразования в зависимости от диапазона представления чисел в системе и заранее заданной точности. Система также позволяет получить расстояние между матрицами целочисленного и классического преобразований. Она может найти свое применение в различных задачах и исследовательских средах связанных с обработкой изображений и сигналов.

Отметим, что в дальнейшем целесообразно использовать обратимые преобразования (reversible transforms) [10], которые дают возможность получать коэффициенты преобразования в том же диапазоне, что и исходный цифровой сигнал.

Литература

- [1] John Miano, "Compressed Image File Format", Addison-Wesley 2003.
- [2] Михаил Гук, "Процессоры Pentium 4 Athlon и Duron", Питер 2001.
- [3] Emmanuel C. Ifeachor, "Digital Signal Processing, A Practical Aproach", Prentice Hall, 2002
- [4] Flores I., "The Logic of Computer Arithmetic". Prentice Hall, 1963
- [5] L. Cheng, Y. Zeng. Fast Multiplierless Approximations of DCT. IEEE Trans. Circuits and Systems-II. 2002.
- [6] Ahmed,Rao. Orthogonal Transforms for Digital Signal Processing. New York. 1975
- [7] S. Oriantera, Y. J. Chen, Truong Nguyen, "Integer Fast Fourier Transform", IEEE 2001.
- [8] K.R. Rao, P.C. Yip, "The transform and data compression handbook", CRC Press 2001
- [9] G.A. Petrosyan, H.G. Sarukhanyan, IDCT-II And Realization Using MMX Technology, Information Technologies and Management. Number 3, 38—43 pages. 2004.
- [10] K.Komatsu and K.Sezaki, "Reversible Subband Coding of Images", Proc. SPIE VCIP, vol.2501, pp.676-684, May 1995.

Ամբողջական ձևափոխությունների սերման ծրագրաշար

Գ. Ա. Պետրոսյան

Ներկայացված է ամբողջական ձևափոխությունների սերման (ITG) ծրագրային համակարգի ընդհանուր նկարագրությունը: Ինչպես հայտնի է ամբողջական ձևափոխությունները աչքի ընկնելով արագությամբ և հիշողության խնայողությամբ որակական ցուցանիշներով գրեթե չեն զիջում նրանց: Ներկայացված փաթեթը հնարավորություն է տալիս ստանալ 2^m կարգի Ֆուրյեի, Սինուսոիդային և Հաարի ամբողջական ձևափոխությունների մատրիցները՝ կախված տվյալ համակարգի քվերի ներկայացման հնարավոր միջակայքից: Ստացված ամբողջական ձևափոխությունները կարող են լիարժեքորեն օգտագործվել պատկերների և ազդանշանների մշակման խնդիրներում: